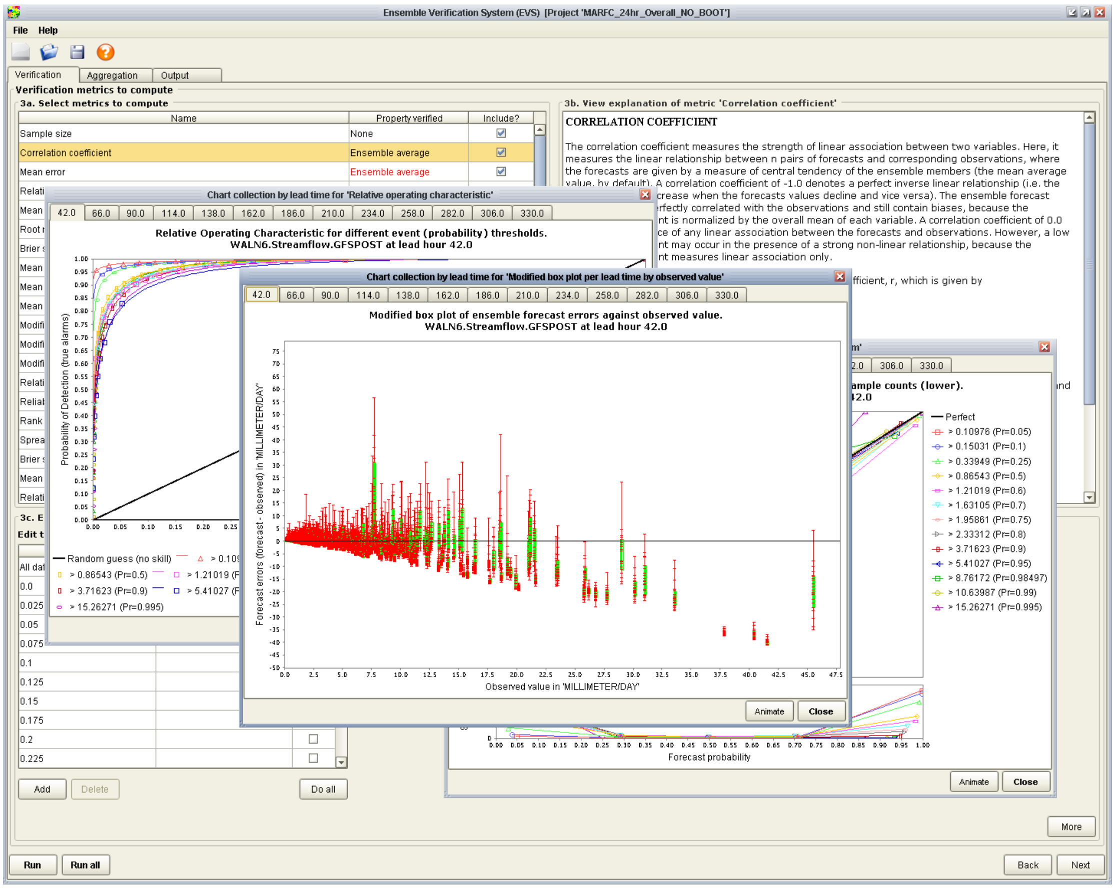
Ensemble Verification System (EVS)

Version 5.0



Ensemble Verification System (EVS) [Project 'MARFC_24hr_Overall_NO_BOOT']

Verification metrics to compute

3a. Select metrics to compute

Name	Property verified	Include?
Sample size	None	☑
Correlation coefficient	Ensemble average	☑
Mean error	Ensemble average	☑

3b. View explanation of metric 'Correlation coefficient'

CORRELATION COEFFICIENT

The correlation coefficient measures the strength of linear association between two variables. Here, it measures the linear relationship between n pairs of forecasts and corresponding observations, where the forecasts are given by a measure of central tendency of the ensemble members (the mean average value, by default). A correlation coefficient of -1.0 denotes a perfect inverse linear relationship (i.e. the crease when the forecasts values decline and vice versa). The ensemble forecast rfectly correlated with the observations and still contain biases, because the nt is normalized by the overall mean of each variable. A correlation coefficient of 0.0 ce of any linear association between the forecasts and observations. However, a low nt may occur in the presence of a strong non-linear relationship, because the nt measures linear association only.

fficient, r, which is given by

Release notes

Dr. James D. Brown

Hydrologic Solutions Limited, Southampton, UK.

evs@hydrosolved.com

HSL

Preface

The Ensemble Verification System (EVS) is a Java-based software tool originally developed by the U.S. National Weather Service's Office of Hydrologic Development (OHD) and, subsequently, by Hydrologic Solutions Limited, Southampton, UK. The EVS is designed to verify ensemble forecasts of hydrologic and hydrometeorological variables, such as temperature, precipitation, streamflow and river stage. The software is intended to be flexible, modular, and open to accommodate enhancements and additions, not only by its developers, but also by its users. The EVS is "open source" software and is released under the GNU Lesser General Public License (LGPL), Version 3.0. We welcome your participation in the continuing development of the EVS toward a versatile and standardized tool for ensemble verification.

EVS Primary Point of Contact, evs@hydrosolved.com

Acknowledgments

Parts of this work were funded by NOAA's Advanced Hydrologic Prediction Service (AHPS) and by the Climate Prediction Program for the Americas (CPPA).

Disclaimer

This software and related documentation was originally developed by the National Weather Service (NWS) and, subsequently, by Hydrologic Solutions Limited (HSL), hereafter referred to as “The Developers.” Pursuant to title 17, section 105 of the United States Code this software is not subject to copyright protection and may be used, copied, modified, and distributed without fee or cost. Parties who develop software incorporating predominantly NWS developed software must include notice, as required by Title 17, Section 403 of the United States Code. The Developers provide no warranty, expressed or implied, as to the correctness of the furnished software or its suitability for any purpose. The Developers assume no responsibility, whatsoever, for its use by other parties, about its quality, reliability, or any other characteristic. The Developers may change this software to meet their own needs or discontinue its use without prior notice. The Developers cannot assist users without prior agreement and are not obligated to fix reported problems. The EVS is released under the GNU Lesser General Public License (LGPL) Version 3.0. A copy of the LGPL is provided with this distribution.

Contents

1 Changes from EVS 4.0 (4.0.1-0.3.1) to EVS 5.0 (5.0.1-0.3.2)

1.1 Changes in default behavior

- Altered the default functionality of the EVS when reading ASCII files to interpret the date strings in a given format without leniency. For example, 200005 was leniently processed when the required format was yyyyMMddHH, despite the missing ddHH. Due to the Java API for Calendar, this lead to an undesirable interpretation of the dates (producing wrong dates without warning) when the dates on file were corrupt, instead of failing (desirable). A non-lenient interpretation of dates is now implemented and an error thrown if the dates on file do not match the required date format.
- When deriving climatological probabilities from a specified source of observed data, the climatological probabilities are derived from the paired observations unless explicitly requested to use all available observations (from the GUI, using the advanced option: 2b. Identify input data sources > More > Other options > Use all observations (not just paired) to determine climate thresholds; and from the EVS project file using the XML tag <use_all_observations_for_climatology>). Previously, the climatological probabilities were determined from all available observations *between the start and end dates of the verification period*, without any options to control this behavior. Now, the climatological probabilities may be determined from the paired data, from all available observations, or from an arbitrary subset of observations (by controlling how date and value conditions are applied to the observed data).

1.2 Bug fixes related to Graphical User's Interface

- Updated the box plots by observed value to include the units of measurement on the domain axis (previously this was stated on the range axis only).
- In the second window of the Verification stage, clicking "Do all" to apply the thresholds for the currently selected verification metrics across all other metrics required the thresholds in the table to be finalized (i.e. the table row to have stopped editing) otherwise erroneous thresholds were reproduced across all metrics. The editing is now finalized automatically before the thresholds are copied.

- Fixed a bug in AggregationA.java whereby the AUs became incorrectly ordered upon saving, leading to an inconsistent state in the locally-stored parameter values.
- In the deleteReferenceFcst method of VerificationA.java, a map iterator was modifying elements in a map concurrently, leading to a ConcurrentModificationException.
- In VerificationB.java, when identifying thresholds as "main" in the threshold table, a modification followed by a save was not being registered on reloading a project because the table row selection was maintained across loading projects, yet the save index was cleared, leading to a conflict between the two.
- The RankHistogramPlot.java failed to plot the rank histogram in the desired bar chart form when only one “main” threshold was requested. Instead, the bar chart format was only plotted if the threshold count was one.
- Fixed a bug that prevented the proper display of the EVS html help documents in an external browser. The solution involved decompressing the html documentation on-the-fly from the EVS.jar and then pointing the browser to the decompressed files for display. All help documentation can now be viewed on-the-fly in an external web browser.
- Week 53 was not supported as a conditioning option in the GUI (this can occur in leap years). This is now supported.
- The addDefaultUnits method of AggregationA.java failed to add an AU for each unique combination of VUs.
- Fixed a bug in OutputA.java whereby a ClassCastException was thrown upon wrongly casting Double valued lead times in the lead time table as Integer objects. This only occurred when selecting a subset of lead times for display.
- The results from all VUs were removed upon cancelling the calculation of one verification unit (i.e. the clean-up extended too far, removing all results).

1.3 Bug fixes not related to Graphical User's Interface

- Sample counts were not being written properly to the XML metadata files.
- Timezone offsets other than UTC (i.e. zero offset) in the PI-XML forecast files were not being processed correctly.
- Fixed a bug in method setTargetMeasurementUnitsFunc of evs.analysisunits.scale.Support, whereby a request for no change in measurement units could be coupled with a function that *did* imply a change in

measurement units, leading to an undesirable change in units. Added a check to prevent this.

- When using another VU as a reference forecast in a skill calculation, any renaming of that VU was not properly reflected in the list of reference forecasts.
- When loading AUs into the GUI, the project file loaded without warning if the AUs comprised VUs that were miss-spelled. Modified a try/catch block to throw an appropriate error, resulting in the (correct) failure to load the project.
- Fixed a bug in the loading of project files whereby the display of aggregation units was not updated if an existing project was loaded or existing units were defined but not saved in a project.
- The readInHeader method of the evs.data.fileio.ohdfile.data.Datacard file reader incorrectly threw a NullPointerException rather than an IOException when reading a null file handle, leading to an ambiguous error message about the cause of the problem.
- Fixed a bug in the readAndSetClimatology method of PairedDataSource.java whereby user-specific IDs for the observed data other than locationID and variableID were not being checked and used if available. For the PI-XML data, several IDs may be used at any given location, with different IDs for observations and forecasts, hence the need for user-specific file IDs separate from the main IDs used for verification.
- In the ProjectFileIO.java class, relative paths were not correctly saved either for aggregation units or for the paired data. Relative paths can now be used for all data sources in the EVS. Paths are resolved relative to the directory in which the EVS.jar is located.
- In the ASCIIFileIO.java, date strings of different length than the specified format string were still processed (i.e. the dates were interpreted leniently), leading to unexpected dates upon reading ASCII files. A non-lenient implementation was adopted.
- Fixed a bug in the determination of verification thresholds from the climatological observations. While a file containing an extended, unconditional, climatology could be specified in place of the paired observations, the unconditional climatology was incorrectly ignored under some circumstances.
- In VerificationUnit.java, a change of units to an unconditional climatology was applied to missing data (-999.0). Following application to the missing data, these data were no longer interpreted as missing but (incorrectly) as non-missing data with arbitrary values (depending on the requested change of units).

- Fixed a bug in the PISAXHandler.Java whereby a decimal time offset in XML was not being read and interpreted properly, leading to an (incorrect) time offset from UTC of zero. This occurred when the time system in PI file was mistakenly expressed as a decimal (e.g. -5.0 for EST). A partial fix to this bug was implemented previously, but the fix failed under specific circumstances, now corrected.
- In setting explicit forecast/observed file IDs for PI-XML etc., the check on equality of null IDs was failing, leading to existing pairs and results being deleted upon opening of an additional options dialog (that contained the IDs) after a verification run; paired data are always auto-deleted upon a change in a parameter that potentially affects the pairs. The bug/improper method invocation was located in StringUtilities.nullEquals. This method was enhanced to take an additional parameter allowing empty strings to be treated as null.
- Fixed a bug in the specification of weeks of the year on which to condition, whereby integer weeks of the year were interpreted as one week too late.
- Fixed a convoluted complex bug whereby AUs were not being deleted properly for VUs whose parameters had changed, dictating the deletion of the AU. This was due to the deletion taking place within the loop in which VUs were checked, thus allowing a VU to be associated with an AU that would later disappear due to changes in other VUs (later in the loop). This specifically occurred when the offending VU comprised an AU with only two VUs of which the earlier VU was one. Moved the check for setting AUs outside of the loop for checking whether AUs should be deleted.
- In the temporal aggregation routine getTimeAggData, the number of rows to aggregate was determined from the separation in time between two forecasts. However, when a time constraint on the processed data led to the first trace having a single time, the hour determination between forecasts was incorrect. The bug was fixed by splitting the data into traces and finding the first one with more than two consecutive times from which the forecast frequency could be determined correctly.
- The read method in PublishedInterfaceXMLIO.Java threw a generic IOException, yet code in the PairedDataSource.java that calls this method relied on a ConditionException being thrown and caught when the file was processed properly but did not contain data of interest. The result was that, when processing data for the reference forecast only, the appropriate exception was not being caught and thrown and the reading of other files (with valid data) did not continue.

Instead, the whole I/O failed indicating that no data could be read for the reference forecast, due to no data being accepted from a single file. The code in PublishedInterfaceXMLIO.Java was updated to throw a ConditionException.

- Fixed a bug in the identification of VUs to include in AUs whereby the VerificationUnit.aggregationEquals method was testing the quality of the support for the forecasts and observations using the equals method (exact equality) rather than the evs.analysunits.scale.TemporalSupport,equalsOnUnitChange method (equality) after any requested changes in measurement units. This led to valid candidates being rejected for having different support, despite the equivalent support after the requested unit change was implemented.
- A memory leak was discovered in the EVS when using a large number of stations (e.g. > 100). This stemmed from a persistent reference to paired data within each VU, after the VU had been processed. The logic for retaining the pointer was to allow for quicker calculations when subsequently processed VUs depended on earlier units (e.g. for skill scores or aggregations across locations). The code was improved to release the pointer to old pairs when no longer needed.

1.4 Feature upgrades and modifications related to Graphical User's Interface

- Re-labeled the "Start" button in the GUI Output dialog to read "First." Previously, “Start” was confused to mean generate output (“Run”).
- Removed the individual "Save" buttons from each GUI window, as the "Save" option is always accessible from the task bar. This avoids confusion about when to save a project.
- When exiting the GUI, the options now include saving the current project, not saving the project, or cancelling. Previously, only exit (without save) and cancel options were available, potentially leading to the inadvertent failure to save a current project.
- Added warning messages on setting the date formats for ASCII whereby case sensitivity in the date formatting could be missed, inadvertently, leading to the incorrect interpretation of dates (e.g. m versus M for minute versus month). This required additional warnings in AnalysisUnit.java, ASCIIFileIO.java, OHDFileIO.java and a new method in StringUtilities.java to check the dates.
- Added several options to the MoreVerificationWindowDialog.java to allow for conditioning on dates that represent forecast issue times as well as valid times.

- Added parameter support and plotting options for a new verification metric, namely the rank histogram.

1.5 Feature upgrades and modifications not related to Graphical User's Interface

- Allowed for conditioning of verification pairs with date conditions that comprise forecast issue times as well as valid times.
- Added the rank histogram to the available verification metrics in the EVS. Tied ensemble members are treated by randomly sampling from the tied ranks. By default, the rank histogram comprises the relative frequencies (probabilities) that an observation falls between two ensemble members, in order to allow simpler comparisons between thresholds or forecast datasets with different sample counts. However, an advanced option allows for the sample counts to be expressed in absolute units rather than empirical probabilities. When the rank histogram is computed for a single dataset, without the use of thresholds to subset data (i.e. conditional rank histograms), a traditional histogram is used. When multiple subsets of the original sample data are used (i.e. thresholds), the resulting “conditional rank histograms” are plotted with lines for improved visibility.
- Allowed for simple transformations of the EVS XML outputs using an Extensible Stylesheet Language Transformation (XSLT). The transformation is implemented on the command line using a XSLT style sheet. This simplifies the use of the EVS outputs in secondary applications, which require a different format than the native format output by the EVS.
- Added a minimum sample size constraint when computing metrics. If the sample size falls below the minimum constraint, the metric is not computed. The precise meaning of that constraint varies between metrics. For example, when computing dichotomous metrics (i.e. metrics computed with respect to discrete events), such as the Brier Score or reliability diagram, the minimum sample size refers to the smaller of the number of occurrences and non-occurrences of the event. For continuous metrics, such as the mean error or the CRPS, it simply comprises the number of (possibly conditional) samples from which the metric was computed. This option is only available via the EVS project file using the XML tag <minimum_sample_size_parameter> for a chosen metric, together with an integer value for the minimum sample size.
- Allowed custom attribute units to be defined (previously, this was constrained to units in the EVS unit library).

- Expanded the default attribute units available in the EVS unit library.
- When climatological thresholds are specified that refer to duplicate real-valued thresholds, these duplicates are no longer computed.
- Updated PairedData.java to store verifying observations locally in order to avoid re-reading them from file. Also, separated between the verifying observations read from file (the unconditional observations) and those associated with the paired data. This allows for an unconditional climatology to be stored locally also. An immediate benefit was a reduction in the run times when conducting verification with large observed datasets.
- In order to circumvent the high memory usage by the EVS for those forecast locations where the forecasts are much more resolved (e.g. hourly) than the temporal resolution required for verification (e.g. daily), an option was included to store the paired data in their aggregate (e.g. daily) resolution rather than their native resolution. Further, the forecast I/O was updated to allow for "on-the-fly" aggregation of the forecasts (i.e. after reading each file), thereby avoiding the need to read and store all forecasts at their native resolution.
- Reading of NWS binary files for "Conditional Simulations" (CS) was previously restricted to forecast data, not observations, yet the NWS binary files may contain simulated streamflows that are useful for verification purposes. Of the NWS formats previously supported, only CARD was supported for observations and NWS binary for forecasts. The NWS binary (CS) format is now supported for verifying observations (which may comprise simulated streamflows).
- Expanded the range of options for allowing conditioning of the sample climatology in the EVS as a baseline forecast (i.e. the use of a "conditional climatology" as the baseline).
- Added new constraints for restricting the reading of forecasts and observations when conducting pairing. Previously, limited constraints were provided on a file-reader-specific basis for restricting the range of dates and lead times processed. This was abstracted into a separate class, ConditionArray.java, for storing an array of conditions. Constrained I/O is now implemented consistently across all file readers, simplifying the coding of further constraints in future. Currently, constraints are allowed on the period of record and on forecast lead times.
- Improved the memory management during pairing of very large datasets by PairedData.java (e.g. required for long-range hindcasting). Specifically, updated the sortTraceByTrace method that's sorts the pairs in trace order, using a new ForecastTime.java object to automatically sort the pairs in a TreeMap, rather than

using layered TreeMaps. The resulting code is marginally slower for small datasets, but significantly improves the memory management for very large datasets.

- Added several new temporal aggregation functions to the FunctionLibrary.java class.
- Abstracted an option to conduct strict temporal aggregation (ignoring data blocks with any missing data versus processing the non-missing data) from a class variable in PairedData.java to a method variable in getTimeAggData, allowing strict aggregation to be controlled via the appropriate method call, rather than being hard-coded to always perform strict aggregation.
- Added functionality in PairedData.java to handle NaN values, as well as missing values, and also updated the missing value identifier in VerificationUnit.java to ensure that NaN is not a valid missing value identifier. In the process, abstracted the search for missing and NaN values to more generic functions in FunctionLibrary.java, an example of which is the new VectorDoubleProcedure.java
- Added an option to specify the graphics output format on the command line by adding a graphics format tag underneath the <output_data> tag in the project file, namely <output_graphics_format>, with supported options comprising JPEG, PNG and SVG (Scalable Vector Graphics) and the default, JPEG.

1.6 Feature upgrades and modifications for developers

- Moved all non-source files from the /src directories to /nonsrc. This required some code improvements to locate parameter files that were previously located underneath the source tree. The refactored structure is cleaner, ensuring only source code is located underneath the source tree.

2 Changes from EVS 3.0 (build 10/01/10) to EVS 4.0 (build 10/07/11)

2.1 Changes in default behavior

- The observed and forecast file types must now be explicitly defined. This avoids ambiguities and performance issues that can arise from auto-detecting the file types (by opening and reading in the first few bytes in the file and checking against expected types). The default data types are ASCII. Thus, old project files for which paired data are no longer available will encounter an I/O error message if the file types are not ASCII (requiring the types to be explicitly defined for the first time).

- Changed the metric type of the ROC Score from a test of the “Ensemble distribution” to a test of “Ensemble skill.” Arbitrary reference forecasts are now supported for the ROC Score, not just sample climatology.

2.2 Bug fixes related to the Graphical User’s Interface

- Corrected several broken hyperlinks in the html descriptions of the verification metrics, which are displayed in the second window of the “Verification” stage.
- Fixed a bug whereby the internal hyperlinks between the descriptions of the EVS verification metrics (i.e. from one html description to another) were not accessible. This was due to the packaging of these html descriptions within a single executable JAR file. In order for the internal links to work properly, a separate directory containing the html descriptions (outside of the executable JAR file) is required, and is now provided with the distribution, namely /evs/resources/statsexplained. The consequences of moving the executable JAR or not downloading this separate directory are minor (i.e. the internal hyperlinks will not work, and navigation to the appropriate description must be performed manually). However, an error message is now provided in that case, indicating that the html resource cannot be found locally.

2.3 Bug fixes not related to Graphical User’s Interface

- Corrected the importing of NWS Card files to account for several forecast issue times within one day. Previously, the issue date was represented by the year, month, and day of month, but not the hour of day. This led to the duplication (and subsequent elimination) of verification pairs that were assumed, incorrectly, to originate from the same hour of the day.
- Fixed several bugs in the reading of PI-XML files, including the failure to read observations in PI-XML format.
- Fixed several minor bugs identified by FindBugs: http://findbugs.sourceforge.net/

2.4 Feature upgrades and modifications related to Graphical User's Interface

- Added the facility to specify the observed and forecast file types in the first verification window, via drop-down menus.
- Added the option to manually specify the date formats for ASCII observed and forecast files. These options appear in the "Other options" pane of the "Additional options" dialog. The "Additional options" dialog is accessed using the "More" button associated with the input data (2b) in the first window of the "Verification" stage.
- Added an option to specify the number of decimal places for writing verification pairs (the default is 5). This option appears in the "Other options" pane of the "Additional options" dialog. The "Additional options" dialog is accessed using the "More" button associated with the input data (2b) in the first window of the "Verification" stage.
- Added an option to use all observations when computing the sample climatology corresponding to real-valued thresholds. This option appears in the "Other options" pane of the "Additional options" dialog. The "Additional options" dialog is accessed using the "More" button associated with the input data (2b) in the first window of the "Verification" stage.
- Added an option to suppress the writing of conditional pairs via the GUI. Previously, this option was available in the project file only. It is now accessed using the "More" button associated with the output data (2d) in the first window of the "Verification" stage.
- Added the RME and associated explanation and parameter options to the second window of the "Verification" stage.
- Added an option to apply the thresholds defined for a selected verification metric to all other metrics associated with the current Verification Unit. The IDs and status of the thresholds as "Main" (i.e. thresholds for display) are also copied. When copying thresholds that comprise an unconditional constraint (i.e. use of "All data"), the unconditional threshold is only copied to metrics that are based on discrete events. The feature is accessed via the "Do all" button under the basic parameter options (3c) in the second window of the "Verification" stage (after selecting a metric that comprises thresholds as a basic parameter option).
- Reorganized and improved the display of additional parameter options for the verification metrics. These are accessed via the "More" button from the basic parameter options (3c) in the second window of the "Verification" stage.

- Added options relating to the calculation of confidence intervals for the verification metrics. These include specifying one or more intervals in the range [0,1], together with the parameters of the stationary block bootstrap algorithm. These options are included in a separate tabbed pane (labeled “Confidence intervals”) under the advanced parameter options for a particular metric. The advanced parameter options are accessed via the “More” button from the basic parameter options (3c) in the second window of the “Verification” stage. Additional options apply to the calculation of confidence intervals for Aggregation Units. These options are accessible via the “More” button, which is located adjacent to the tabulated list of Verification Units (2b) in the “Aggregation” stage. The options include omitting the calculation of confidence intervals for the Aggregation Unit (regardless of whether they are required for particular Verification Units) and specifying whether the component Verification Units are statistically dependent.
- Added the option to pool verification pairs across several forecast locations and to compute the verification metrics from the pooled pairs, rather than averaging the metrics from the individual locations. Previously, this option was only accessible via the XML project file. It is now accessible via the “More” button, which is located adjacent to the tabulated list of Verification Units (2b) in the “Aggregation” stage.
- Added shortcuts for selecting particular combinations of metrics in the products table (1b) of the “Output” stage. These options are accessed by right-clicking on the products table. New options include the ability to select all forecast lead times for the selected metrics across all Verification Units (e.g. to output results for the correlation coefficient across several Verification Units).
- Improved the error handling and reporting code, providing more specific and detailed error messages.

2.5 Feature upgrades and modifications not related to Graphical User's Interface

- Added the facility to manually specify the date formats used in the ASCII observed and forecast files. A date format is based on the elementary components, yyyy (year), MM (calendar month), dd (day of month), HH (hour of day in the 24-hour clock), mm (minute of hour) and ss (second of minute). The elements are separated with single characters or whitespace (e.g. MM/dd/yyyy

HH) or appended without separators (e.g. yyyyMMddHH). The default date format is MM/dd/yyyy HH.

- Added the relative mean error (RME) to the set of deterministic metrics available in the EVS. The RME comprises the mean error as a fraction of the mean observed value over the sample.
- Added an additional method for computing the Area Under the Curve (AUC) for the empirical Relative Operating Characteristic (ROC) Score. The default method remains the algorithm described in Mason and Graham (2002): Mason, S.J. and Graham N.E., 2002: Areas beneath the relative operating characteristics (ROC) and relative operating levels (ROL) curves: Statistical significance and interpretation, *Quarterly Journal of the Royal Meteorological Society*, **30**, 291-303. The alternative involves computing the AUC from the empirical ROC curve, which is based on a finite number of points/classifiers. The integral of the empirical ROC curve (AUC) is computed using the trapezoid rule. In most cases, the algorithm described by Mason and Graham (2002) generates larger values of the AUC (skill) than the integral of the empirical ROC curve.
- Extended the ROC Score to allow for arbitrary reference forecasts, not just sample climatology.
- Added the likelihood-base-rate (LBR) decomposition of the Brier Score into Type-II conditional bias, discrimination and sharpness, and the corresponding decomposition of the Brier Skill Score into relative Type-II conditional bias, relative discrimination and relative sharpness.
- Added options for computing confidence intervals for the verification metrics (except the box plots). One or more confidence intervals may be computed for selected metrics of a Verification Unit and for an Aggregation Unit. The confidence intervals are derived from a stationary block bootstrap of the verification pairs. In order to account for temporal statistical dependence, the bootstrap resampling applies to contiguous “blocks” of pairs rather than individual pairs. The blocks are parameterized by their mean length and are sampled from a geometric distribution with that parameter. In order to account for spatial dependence (when computing aggregate verification results across several locations), the absolute times of the sampled blocks may be coordinated/fixed across all locations. The bootstrap algorithm is multi-threaded for improved performance on multi-core, multi-processor, machines.
- Provided additional command line options for suppressing the writing of either the graphical or numerical outputs when running the EVS in batch mode. The writing

of graphical outputs is suppressed with –g and the writing of numerical outputs is suppressed with –n.

- Provided additional command line options for converting between legacy file formats (NWS Card and NWS CS binary) and a generic ASCII file format used by the EVS. The option –bin2asc in.cs out.fcst converts the NWS CS binary file, in.cs, to the ASCII output file, out.fst. The option –fcard2asc in.fcst out.fcst converts the forecast data in NWS Card file, in.fcst, to the ASCII output file, out.fst. The option –ocard2asc in.obs out.obs converts the observed data in NWS Card file, in.obs, to the ASCII output file, out.obs.
- Improved the R utilities script for reading the XML numerical outputs from the EVS into R (www.r-project.org). The utilities script is located in /evs/resources/rscripts/Utilities.R. There are three key methods for reading the different EVS outputs, namely readEVSScores, which reads the verification scores, readEVSDiagrams, which reads the verification diagrams (e.g. reliability diagram, spread-bias diagram) and readEVSBoxPlots, which reads the EVS box plots.
- Provided two self-contained R scripts in /evs/resources/rscripts/example_scripts to demonstrate the plotting of EVS output in R. The necessary EVS (XML) outputs are located in evs/resources/rscripts/example_scripts/example_evs_out.
- Added an option to omit 'no-data' values from the paired files.

2.6 Feature upgrades and modifications for developers

- Abstracted control of metric calculation by lead time from the individual metrics to a new method in the evs.metric.metrics package, namely computeByLeadTime. This leads to much cleaner and more extensible code. For example, it is no longer necessary to iterate through lead-times when implementing the compute method of a new metric. Rather, the compute method is now generic for the given input data. There are several downstream effects of these changes that have led to significantly better performance, particularly when computing bootstrap confidence intervals (i.e. repeatedly calculating the metrics).

3. Changes from EVS 2.0 (build 10/01/09) to EVS 3.0 (build 10/01/10)

3.1 Changes in default behavior

- Changed the temporal aggregation default to store the maximum valid time of the input times (in UTC) when conducting aggregation. Previously, the default was to compute the mean of the input times. The default for handling the forecast lead times remains to compute the maximum of the inputs. Thus, for example, aggregation of four six-hourly pairs at increasing UTC times of {18, 0, 6, 12} previously generated an aggregated paired value with time UTC 3, but will now generate an aggregated paired value with time UTC 12 (note that 12 proceeds 18 when considering date). Thus, the aggregated value should be interpreted as the value over the period of aggregation immediately preceding the stated time.
- Changed the start and end dates of the verification period defined in the first verification window from the forecast time zone to UTC. The start and end dates begin and end at 00 UTC on the specified date, respectively. Thus, in order to include verification pairs that fall on the specified end day, one day should be added to the input date.
- Changed the order of error messages displayed in the GUI Console window (not to be confused with an external console) so that the latest error messages are displayed at the top of the console rather than appended to the bottom.

3.2 Bug fixes related to Graphical User's Interface

- Fixed a bug in the table of thresholds for each verification metric, ensuring that the scrolling window expands properly as new thresholds are added (previously a fixed limit).
- Fixed a bug in the table of candidate units for aggregation, ensuring that the scrolling window expands properly to show all available units (previously a fixed limit).
- Corrected a bug in the GUI for selecting pre-conditions to apply to the verification pairs. Entering incorrect conditions lead to a (correct) warning, but, when subsequently cancelling further edits, the existing (valid) conditions were removed rather than being returned to the original (valid) state.

3.3 Bug fixes not related to Graphical User's Interface

- Fixed a bug in the BSS and CRPSS to write the null value identifier where the output of the BSS or CRPSS is undefined (i.e. divide by zero), thereby allowing proper display in the plots for those lead times where the score *is* defined.
- Fixed a bug in the reading of an XML paired file where one or more ensemble members were missing, i.e. the number of members in the paired file (which does not store missing members) varied. The result of this bug was that missing members were initialized with the java default value for a float, namely 0.0, and not the pre-defined null value identifier.
- Fixed a bug in the skill score calculations (BSS and CRPSS). When these calculations were performed repeatedly (i.e. by clicking “Run” two or more times), the skill scores were zeroed on the second or further runs. While the metrics would not normally be re-computed, the bug has been fixed.
- Fixed a bug in the application of value conditions, whereby the conditions were being applied *before* any requested changes in measurement units rather than *after*. The value conditions (and hence real-valued thresholds on which those conditions are based) are now in the target measurement units.
- Fixed a bug in a method for ordering the paired data by trace, whereby the last trace (in order of forecast valid time) was not being appended to the results and, therefore, included in the verification results.
- Spell-checked all developer documentation and corrected spelling mistakes.

3.4 Feature upgrades and modifications not related to Graphical User’s Interface

- Packaged the EVS into a single executable JAR file with all associated libraries using an ANT build script. The EVS is now delivered as a single executable JAR file, without the need to maintain an internal directory structure for dependent libraries (which are now packaged and accessed from within the executable JAR).
- The writing of conditional pairs (i.e. a subset of the overall pairs with any conditions on variable value or date applied) has been made optional to speed-up the processing of large numbers of verification points. This option is implemented via the <write_conditional_pairs> tag of the <paired_data> section of the EVS project file, with a default of true, i.e. conditional pairs are written by default, as before. This option is also accessible via the GUI (see below).

- Improved the performance of the temporal aggregation routine and provided options for the type of aggregation function applied not only to the forecast data, but to the forecast valid times and lead times. Also changed the default behavior; previously, the forecast valid time was given the mean of the input times and the lead time assumed the maximum of the input times; the forecast valid time now assumes the maximum of the input times (see below).
- Improved information and error messages printed to standard out (i.e. the console, if EVS was initiated from a console window).
- Added options for aggregating the support of the observed data to match the support of the forecasts, including the ability to compute an accumulation over a forecast window. Eventually, an aggregation function will be implemented for every possible combination of input support allowed in the EVS (including a change of measurement units).
- Added methods for reading paired data in the same ASCII format to which the XML pairs can be converted. Thus, paired files may be produced and read in ASCII format (as well as forecasts and observations) where convenient.
- Implemented additional R scripts for plotting the verification results produced by the EVS (i.e. the XML output), including a script that will plot the EVS single-valued metrics and ensemble scores as a function of threshold value.
- Implemented the three-part decomposition of the Brier Score into: Brier Score = reliability – resolution + uncertainty and the associated graphical and numerical products.
- Implemented additional options for averaging the forecast ensemble members when computing single-valued verification metrics such as the mean error, root mean square error, mean absolute error, and correlation coefficient. The default remains to compute the ensemble mean. Additional options now include the ensemble median and mode.
- Added the Mean Absolute Error of the ensemble average to the single-valued verification metrics.
- Added the climatological frequency and the zero-skill line to the reliability diagram (located half-way between the climatological frequency and the expected frequency for a reliable forecasting system). These curves are available in the XML output files, but are not plotted within the EVS.
- Backwards compatibility has been maintained for earlier project files. Upon saving old project files within a new version of EVS, new options will be written with their default values.

- Implemented the concept of “main” and “auxiliary” thresholds for metrics that either require or support thresholds. Currently, the “main” thresholds are used to identify events (or subsets of data) that should be included in the graphical outputs from the EVS. By default, both the “main” and “auxiliary” thresholds are included in the numerical outputs from the EVS. This information is stored as an additional XML tag in the project file, labeled main_threshold, which contains a list of Boolean values equal in length to the number of thresholds (true indicates a main threshold).
- Added the facility to derive climatological probability thresholds from a larger set of observed data. In the EVS Version 2.0 they were derived from the paired observations, after applying any requested changes in units, temporal aggregation, or value and date conditions, including the discrete verification time-period requested in the first verification window. Now, they may be derived from the original observed data, again after applying any requested changes in units, temporal aggregation, or date and (observed) value conditions, but NOT the discrete verification time-period (i.e. the full period of record covered by the observed file will be used, after applying any changes in measurement units, temporal aggregation, value conditions on the observations, and date conditions *except* for the reduced verification time period). This option is controlled by a Boolean flag, which is accessible via a check box in the GUI, and also in the project file under the new XML tag, labeled use_all_observations_for_climatology. Note that this option only applies to the derivation of real-valued thresholds corresponding to particular climatological probabilities of occurrence. For those metrics that incorporate climatological probabilities in the calculation (e.g. the climatological frequency in the reliability diagram), the behavior is unchanged (the observations associated with the conditional pairs are still used).
- Added further date and value (pre-)conditions, including additional statistics for selecting forecasts based on value (ensemble median and mode, probability of not exceeding a given value, and the value corresponding to a given non-exceedence probability) and additional functions for selecting forecasts and observations based on dates (hours of day in UTC). Thus, much more complex pre-conditions are now possible, such as selecting only those pairs (for verification) whose forecast probability of exceeding “flood stage” is greater than 0.9. This functionality is necessary for real-time verification, where the aim is to

select historical (observed and forecast) analogs to a real-time forecast based on specific properties of that real-time forecast, and possibly auxiliary information.

- Added the facility to compute the binormal approximation to the Relative Operating Characteristic (ROC) curve and the associated ROC Score. The approximation is based on fitting the binormal model to the empirical (POD, POFD) pairs and is, therefore, dependent on the number of thresholds chosen. For an exact comparison between the binormal approximation to the ROC curve and the binormal approximation to the ROC Score, a common number of thresholds should be adopted for each metric. However, when comparing the empirical ROC Score to the binormal ROC Score, the results will be closest when adopting *m*+1 thresholds, where *m* is the number of ensemble members per forecast. Specifically, the empirical ROC Score is derived from ranking of the POD and POFD data, rather than computing the ROC curve. The ranked data can take at most *m*, “jumps” in probability (at the corresponding ensemble member positions). Hence, the empirical ROC Score is analogous to deriving the ROC Score from an empirical ROC curve constructed with *m*+1 thresholds.
- Added the facility to aggregate the observed support prior to verification. Previously, this was only possible for the forecasts. The same restrictions apply to change of support of the observations as the forecasts, namely the aggregated support is exactly divisible by the frequency of the data and comprises either a mean of the input values if the inputs have instantaneous support or a total of the input values if the input values are totals.
- Added the option to remove certain lead times from the verification results based on a minimum sample-size requirement. The sample size constraint is set by a fraction in the range [0,1]. The fraction is multiplied by the average number of pairs across all lead times to determine the minimum sample size in numbers of pairs. For example, a fraction of 0.5 implies that verification results will *not* be computed for any lead time with fewer than 50% of the average number of pairs across all lead times.

The following new features are only accessible via the EVS project file (not the GUI):

- Added the facility to specify the method for computing CRPS in the EVS project file. By default the Hersbach (2000) method is used, but a method that can handle null ensemble members has been added. This is specified in the <crps_method> tag of the crps metric in the EVS project file, with options hersbach and with_nulls. If null members are present, the hersbach option will lead to all forecasts with one or more null members being removed from the calculation, and will fail to compute if all forecasts contain one or more null members.

- The facility to prevent elimination of duplicate pairs (pairs with a common forecast valid time and lead time) has been added to the EVS project file. This is implemented via the <eliminate_duplicates> tag of the <paired_data> section of the EVS project file, with a default value of true. When false, duplicate pairs will not be eliminated. This is necessary when computing verification metrics for data that have been pre-pooled across several forecast locations and contained in a *single* paired file.
- Added an option to pool the verification pairs across several forecast locations and to compute the verification metrics from the pooled pairs, rather than averaging the metrics from the individual locations. Theoretically, this approach is preferred, but is much more time-consuming in practice, and is usually not feasible. The default behavior remains to average the verification metrics from the individual locations. The new option is only accessible via the EVS project file by adding or setting the <pool_pairs>true</pool_pairs> entry to the XML for a particular aggregation unit, where true implies that pooling will be conducted, and false implies averaging.
- Added the facility to change the decimal writing precision of the paired data for a given Verification Unit. This applies to both the raw and conditional pairs. The default behavior is to write forecasts and observations with a maximum precision of five decimal places (unchanged), with fewer places written as required. The integer number of decimal places (>0) can now be defined in the project file using the <paired_write_precision>5</paired_write_precision> tag, which is part of the <paired_data> block. This functionality is not accessible via the GUI, and existing pairs will not be re-written with a new decimal precision (unless re-writing is otherwise necessary).
- Added the facility to set the behavior for removing null ensemble members when writing the paired file. The default behavior remains to *remove* null ensemble members. This may be changed using the <strip_nulls_from_paired_file>true</strip_nulls_from_paired_file> tag, which is part of the <paired_data> block. This functionality is not accessible via the GUI, and existing pairs will not be re-written (unless otherwise required).

3.5 Feature upgrades and modifications related to Graphical User's Interface

- Simplified the behavior of the reference forecast selection for skill scores. If a skill score is selected and only one possible reference forecast is available, this is automatically selected in the secondary parameters dialog.
- Added a menu to the Verification Units table in the Output window to allow for the (de)selection of all products and lead times for all available units.
- Added advanced options for computing different averages from the ensemble members when using single-valued verification metrics.
- Added auto-recall of the last directory accessed when creating, saving and reading project files so that the last working directory is opened by default.
- Added an option to change the behavior for writing conditional pairs. This is controlled by a check button in the advanced options (accessed via the “More” button) in the “Output” section of the first window in the GUI. The default (slower) behavior is to write conditional pairs.
- Added support for multiple-row selection and deletion in the thresholds table associated with each verification metric (in the second verification window).
- Added the facility to distinguish between “main” and “auxiliary” thresholds for metrics that either require or support thresholds. By default, the “main” thresholds are used for plotting and all thresholds (“main” and “auxiliary”) are added to any numerical outputs written by the EVS.
- Added functionality to generate verification thresholds semi-automatically for verification metrics that either require or support thresholds. The thresholds are generated by entering a number of thresholds, the first threshold value, and a constant increment (positive or negative) between thresholds. This is useful for designing plots that show verification scores as a ‘continuous’ function of threshold value (i.e. outside of the EVS). By default, only those thresholds identified as “main” thresholds are included in the graphical outputs from the EVS, but the numerical outputs (on which custom plots are designed) include all of the thresholds.
- Re-labeled the “Edit no-data value” option in the advanced input data options dialog to the more generic “Edit other options” and added a check box to control the way observed data are used to determine climatological probability thresholds.
- Improved the GUI for selecting pre-conditions to apply to the verification pairs, in keeping with the enhanced functionality for identifying pre-conditions (see above).

- Added option to iconify (or “minimize” in Windows terminology) the EVS GUI while processing a verification project; a button labeled “iconify” has been added to the progress dialog to facilitate this.
- Changed the label Forecast lead period to Forecast lead time horizon in the first window of the GUI.
- Added the binormal approximation to the ROC and ROC Score to the GUI. In both cases, the binormal approximation is appended to the results when selecting to do so under the Advanced Parameter Options dialog. In that case, the empirical ROC data are plotted as open points and the binormal approximation is plotted with a line of the same color.
- Removed the text (in forecast time system) for both the start and end dates of the verification period in the first window of the GUI, reflecting the change to UTC (see above).
- Added an error message when attempting to temporally aggregate forecasts over a longer period than the specified forecast lead time horizon (e.g. attempting to compute monthly averages for forecasts with a lead time horizon of 14 days).
- Improved the error console in the GUI.

3.6 Feature upgrades and modifications for developers

- Added numerous additional methods for developers that assist in sub-setting paired data according to varied conditions met in particular rows or columns of the paired-data matrix. These methods can be found in evs.utilities.matrix.DoubleMatrix2D. The conditions can be made arbitrarily complex by chaining together functions provided in the evs.utilities.mathutil package and applying them to specified rows and/or columns in the paired data matrix.
- Added a method to linearly interpolate observed data to the nearest forecast valid time using a weighed (by temporal separation) combination of the two nearest observations between which the forecast valid time lies. The method is: evs.data.PairedData.linInterpObsToFcsts. This functionality is currently only accessible to developers, but may be included in the GUI in future.
- Enhanced the processing of timing information by adding a dedicated class for representing forecast valid times and lead times: evs.data.ValidTime.

4. Changes from EVS 1.0 (build 05/09/08) to EVS 2.0 (build 10/12/09)

4.1. Feature upgrades and modifications related to Graphical User's Interface

- Removed Time-Series ID and renamed River Segment ID to Location ID.
- Moved basic output options from pop-up window to main Output window.
- Implemented enhanced error dialog with improved error messages.
- Implemented enhanced progress monitor to monitor and display progress of paired-file reading (and included a pair count in the paired file to enable this).
- Updated the aggregation window to include a weighting input in the table of verification metrics. Also renamed some features in this window.
- Added the option to select an arbitrary reference forecast for a skill metric.
- Added the option to show skill score decompositions in a tabbed pane (similar to metrics with one plot per lead time), which may be animated.
- Improved display of zero error line in plots (extended continuously).
- Improved auto-scaling of axes in plots.

4.2. Feature upgrades and modifications not related to Graphical User's Interface

- Added multiplication factor in support dialog to allow simple changes between measurement units (more complex operations, such as a change in temperature units, are not yet supported).
- Implemented reading of PI-XML observations
- Implemented reading of PI-XML forecasts
- Implemented reading of ASCII observations
- Implemented reading of ASCII forecasts
- Changed representation of forecast lead times from integer hours to double-precision float hours to allow verification of forecasts with lead times in fractional hours, thereby extending the EVS to arbitrary forecast lead times.
- Rewrote the online documentation and updated the mathematical formulas for all of the verification metrics.
- Implemented an R script for each metric in the EVS to read in the XML output and produce high quality plots in EPS format for scientific papers.
- Modified calculation of the mean CRPS to account for the relative position of the observation between the two adjacent ensemble members.

- Added ROC score to the available metrics and included a plot by forecast lead time (same as with other scores). The calculation is based on Mason and Graham (2000).
- Added a sample size metric and associated plot to compute the basic sample size information by forecast lead time and threshold. This may be used for exploratory purposes before computing other verification metrics. In future, we may add further metrics for data exploration (of the observed and forecast data rather than the verification pairs).
- Added modified box plot by size of observed value to GUI (previously via the command line only).
- Modified the spatial aggregation routine to compute the expected (mean) value of each metric across a set of Verification Units rather than pooling paired data.
- Included ability to perform a weighted spatial aggregation. The weight is uniform by default and must sum to 1. A non-uniform weight is also permissible and a weight of "S" is used to weight by the sample size at the first lead time (i.e. maintaining constant weights across lead times). If a verification metric is not available for a given lead time or threshold the weights are automatically rescaled to sum to 1, maintaining the correct relative weighting of the available metrics.
- Relaxed constraints on spatial aggregation to allow aggregation for Verification Units with different start and end dates.
- Improved the efficiency of file reading for external file formats to ensure that only data within the specified start and end dates and forecast lead times are fully read (otherwise only the file headers are read to check this information).
- Implemented backwards compatibility for the above features so that they do not prevent running of old EVS projects. However, the aggregation routine and CRPS update has been swapped without the option of backwards compatibility. Thus, old projects with spatial aggregation will produce different results in the EVS 2.0. The CRPS update was a bug fix, voiding the need for backwards compatibility.
- Updated the algorithm for CRPS to the method described in Hersbach, H., 2000: Decomposition of the Continuous Ranked Probability Score for Ensemble Prediction Systems. There are small numerical differences between the old and new algorithms. Also, the procedure described in Hersbach assumes that a constant number of ensemble members is available, whereas the previous method for computing CRPS had no such constraint. Thus, differences will be seen when comparing numbers between systems for which some forecasts comprise null ensemble members.

- Added the decomposition of the CRPS into reliability, resolution and uncertainty, as described in Hersbach (2000).
- Added the Brier Skill Score (BSS) for an arbitrary reference forecast selected by the user.
- Added the Continuous Ranked Probability Skill Score (CRPSS) for an arbitrary reference forecast selected by the user.

5. Changes from EVS 1.0 BETA (build 10/12/07) to EVS 1.0 (build 05/09/08)

5.1 Feature upgrades and modifications related to Graphical User's Interface

- Removed the table containing reference forecasts, which are not yet enabled.
- Improved the labeling of various options (e.g. 'time zones' rather than 'time systems').

5.2 Feature upgrades and modifications not related to Graphical User's Interface)

- Allowed real-valued thresholds for all metrics.
- Allowed probability thresholds for all metrics, not just Brier, ROC, and Reliability.
- Included the option for thresholding with a closed interval (i.e. a "between" condition).
- Supported the use of symmetric windows around the forecast median in the Spread-Bias plot.
- Included sharpness (sample-count) plot in the Reliability diagram.
- Changed the definition of probability thresholds in Spread-Bias plot, Mean Capture Rate diagram and Box Plots. Previously, these thresholds referred to plotting positions (i.e. plot resolution) and NOT thresholds of the observed distribution. They now refer to thresholds of the observed distribution for consistency with all other metrics. Plotting positions are now determined with a 'points count' parameter. For example, a point count of 10 for the Spread-Bias plot will construct a plot comprising 10 points.
- Added a new 'points count' parameter for the Spread-Bias plot, Mean Capture Rate diagram, Box Plots and Reliability diagram, which allows the resolution of those diagrams to be altered.

- Included the option to change the default temporal aggregation function from the mean over a specified period to one of several other functions, including the minimum, maximum, and total (i.e. accumulation).
- Included units in the plots that comprise real units (mean error, RMSE, Mean Capture Rate diagram, box plots) once those units have been added to the observed and forecast support for a Verification Unit.
- Included an option to animate a sequence of verification graphics at different lead times.
- Included writing of sample counts to an XML file when writing other numerical results.
- Included writing of conditional pairs to XML format as well as the original pairs.
- Included an option to ignore global value conditions on a per-metric basis. For example, if a condition was applied to consider only those pairs whose ensemble mean temperatures exceeded freezing, this condition could be ignored on a per-metric basis.
- Enabled backwards compatibility with old project files (i.e. projects with old options will run as before).
- Enhanced and updated documentation.
- Improved memory management for Aggregation units.